

### AMENDMENTS TO THE SPECIFICATION

#### In the Specification:

- (1) Please replace the paragraph beginning at page 6, line 10 that begins “Many programming languages, for example . . .” with the following amended paragraph:

Many programming languages, for example C# and ~~Java~~ JAVA, catch or prevent many programming error(s) through compile-time checks and/or automatic memory management; however, there remain many programming errors that are not caught until run time, for example: forgetting to release a resource, such as a file or network connection; using a resource after release; calling methods in the wrong order; and/or typographical error(s) in literals that represent dynamic content (*e.g.*, a string that contains a SQL query). Such programming error(s) can cause run-time exception(s), which can be experienced by the software’s customers, if the error(s) are not noticed during testing.

- (2) Please replace the paragraph beginning at page 8, line 3 that begins “For example, the checker 120 . . .” with the following amended paragraph:

For example, the checker 120 can analyze code created in language(s) that compile to the Common Language Runtime (*e.g.*, C#, ~~Visual Basic.net~~ VISUAL BASIC.NET and/or Managed C++). While a modular analysis generally requires extra specification(s), the benefit is that, in one example, the analysis can be performed fast enough to be run after substantially every compilation.